Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information). 8/2 (2015), 87-95
DOI: http://dx.doi.org/10.21609/jiki.v8i2.306

INVESTIGATION OF FLIP-FLOP PERFORMANCE ON DIFFERENT TYPE AND ARCHITECTURE IN SHIFT REGISTER WITH PARALLEL LOAD APPLICATIONS

Dwi M J Purnomo[1], Machmud R Alhamidi[1], Ari Wibisono[1], and M Iqbal Tawakal[4]

[1]Faculty of Computer Science, Universitas Indonesia, Kampus Baru UI, Depok, 16424, Indonesia
[2]KTH Royal Institute of Technology, Stockholm, Sweden

Email: ari.w@cs.ui.ac.id[1]

Abstract

Register is one of the computer components that have a key role in computer organisation. Any kind of computer contains millions of registers that are manifested by flip-flop. This research focused on the investigation of the flip-flop performance based on its type (D, T, S-R, and J-K) and architecture (structural, behavioural, and hybrid). Those type of flip-flop on each architecture would be tested in different bit of shift register with parallel load applications. The experiment criteria that would be assessed are power consumption, resources required, memory required, latency, and efficiency. Based on the experiment, it could be shown that D flip-flop and hybrid architecture showed the best performance in memory required, latency, power consumption, and efficiency. Meanwhile, the greater the register number, the less efficient the system would be.

Keywords: *Memory, flip-flop, performance, type and architecture, performance criteria*

Abstrak

Memori adalah salah satu komponen utama penyusun segala komputer. Memori memiliki jutaan register yang dimanifestasikan oleh flip-flop. Peneitian ini focus pada investigasi ari kinerja flip-flop berdasarkan tipe (D, T, S-R, dan J-K) dan arsitekturnya (structural, behavioural, dan hibrid). Flip-flop dengan tipe-tipe berbeda pada arsitektur masing-masing akan di tes pada aplikasi *shift register with parallel load* dengan jumlah bit yang berbeda-beda. Kriteria yang akan diuji adalah konsumsi daya, sumber daya yang dibutuhkan, memori yang dibutuhkan, latensi, dan efisiensi. Berdasarkan eksperimen yang dilakukan, dapat disimpulkan bahwa kinerja paling baik ditunjukkan pada D flip-flop dan arsi-tektur hybrid pada parameter, memori, latensi, konsumsi daya, dan efisiensi. Sementara itu, semain besar register, semakin tidak efisien sistem.

Kata Kunci: *Memori, flip-flop, kinerja, tipe dan arsitektur, kriteria kinerja*

1. Introduction

To build a computer, regardless the type, the utilization of registers cannot be overlooked. Register grasps key role in the computer organisation, i.e. to store the state and to load it when necessary. One of the foremost component inside the register is flip-flop. Flip-flop is the assembly of several gates, that function to reserve the logical states which is evoked by any data input signal as a response to clock pulses [1]. Flip-flop is employed to receive and store the data sequentially, during predetermi-ned clock interval. The storage is necessary to pur-vey adequate limited time period needed by other components inside the system.

There are various kind of flip flop that exist inside the IC which is provided in the market. The selection of the flip flop to be used depends upon several criteria. Flip flop which exist inside the IC comprising D-Flip Flop, T-Flip Flop, SR-Flip Flop, and JK-Flip Flop.

The first criteria for selecting the suitable flip flop is power consumption [2]. The power is just simply the amount of power which is absorbed by the system. The second criteria is the efficiency of the system and the gate requirement [3]. The effi-ciency of the system is defined to be the ratio bet-ween throughput and used gate. Third criteria is resource necessitated by the system [4, 5]. The re-source needed comprising LUT, slices, and number of flip flop. The difference between resource and gate is that gate is the smaller components that bui-ld resource, for instance in LUT there are several gates which are utilised. Next criteria is memory requirement [6]. Memory requirement is one of the criteria that cannot be overlooked because FPGA only have limited amount of memory. Moreover, memory requirement will indirectly affect the po-wer

required to operate the system. This value depends on the architecture of the system, thus it is highly influenced by the flip flop that is used. Finally, the criteria that has to be taken into account is total CPU time for completion [7]. This value is the delay from the time input was given to the final output is obtained. The seven above-mentioned cri-teria can be utilised to examine and compare one flip flop to the other. There is no best flip flop for every case and criteria, hence optimization is need-ed to design an architecture.

To construct various design and parameters, FPGA is the most suitable device to be used. FPGA can be used to build any circuit without redundan-cy, because the gate contained in the FPGA has not been defined yet. There have been plenty of rese-arch conducted in the architecture comparison uti-lising FPGA as the device. The architecture is de-veloped by employing VHDL language and using miscellaneous circuit design.

Firstly, the research conducted by Panda et al. [7] that implemented binary encoder BCH using VHDL on FPGA. This research focused on the data transferring in AWGN with multiple error correc-tion control. By utilizing simulation and synthesis on FPGA board the various criteria were compared. Secondly, the research conducted by Dondon et al. [8] that investigated the implementation of Artifi-cial Neural Network (ANN) in VHDL on FPGA. This research examined the most efficient archite-cture design for ANN case by means of its speed and resource consumption. Thirdly, research cond-ucted by Lawal et al. [6] that examined memory requirement of real-time video processing on emb-edded FPGA. The paper analyzed the memory re-quirements for real-time video processing on seve-ral FPGA architecture. Finally, paper created by Redif [4] that designed the novel reconfigurable architecture for polynomial matrix multiplication which is implemented on FPGA. This paper conc-erned on the reduction of execution time while li-miting the FPGA resources utilized.

The aforementioned conducted researches, were all focusing on the amelioration of the FPGA architecture. Each of which focused on different case. Nevertheless, the parameters that were inves-tigated to examine the performance of the archi-tecture were similar to the previously mentioned criteria. In this research, those criteria would be compared in the implementation of FPGA to examine the best flip-flop and flip-flop architect-ture. By varying the flip-flop i.e. utilizing D flip-flop, T flip-flop, S-R flip-flop, and J-K flip-flop, the best flip-flop would be figured out. In addition, by varying the architecture by differentiating the approach, the best architecture and the best flip-flop component could be found.

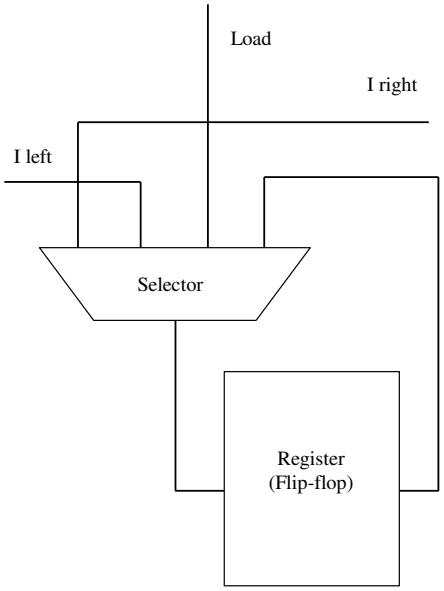

Figure 1. Schematic of 1 bit shift register with parallel load.

2. Methods

In this research, flip-flop performance would be investigated. There are number of test scenarios th-at would be undertaken to asses and compare the performance of each flip-flop. There are four dif-ferent flip-flop type on which the performance test would be imposed, D flip-flop, T flip-flop, S-R flip-flop, and J-K flip-flop. Those four flip-flop would be utilized for similar purpose and would be compared. The application that would be used is Shift Register with Parallel Load. This application was used due to its simplicity and flexibility to be altered. The capacity of this application would be easily changed by altering its bit, thus the con-sumption of its resource could be changed and monitored to give additional scenario for the re-search. Basically, shift register with parallel load is a program which contains four different input and utilizing register to store the data. The inputs are previous output, load input, left side bit of the des-tination bit, and right side bit of the destination bit. The input will be selected based upon the desired operation. Then, the input will enter the flip-flop and processed to obtain output. The schematic of the 1 bit shift register with parallel load is shown in Figure 1.

In this research, the number of bits that would be used is more than one, depend on the capacity of the FPGA utilized. The larger the number of bits, the larger the different of the performance of each flip flip-flop would be. The

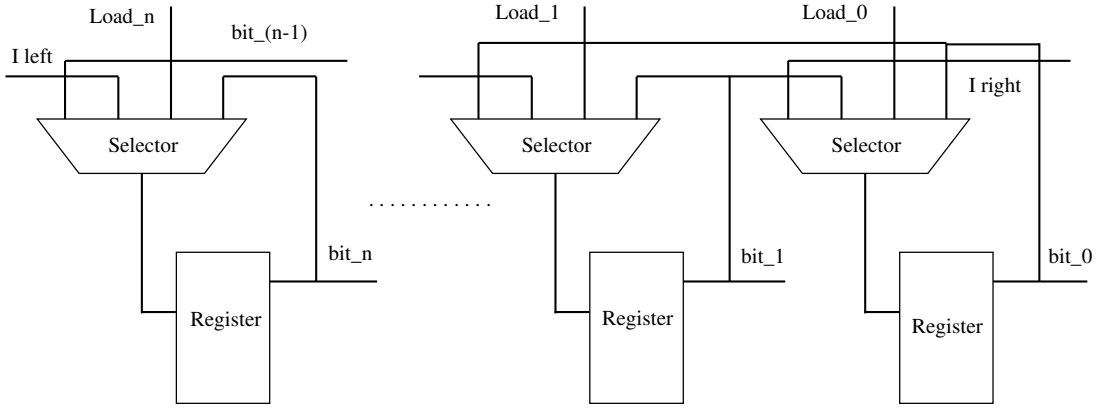

Figure 2. Schematic of n bit shift register with parallel load.

schematic of the n-bit shift register with parallel load is illustrated in Figure 2.

As shown in Figure 2, the input of each bit is related with the other bit. In bit 1 system, the input that will be filled in shift left point is the output of bit 0 system, whereas the input that will be given to the shift right point is the output of bit 2 system. Secondly, in bit 0, the input of shift left point is right input (I right). Meanwhile, the input of shift right point is the output of bit 1 system. Finally, in bit n, the shift left point will be assigned with the output of bit (n-1), and shift right point is filed with left input (I left).

Beside flip-flop type, the architectures of the VHDL code that would be implemented on FPGA were also varied. There are three different architectures that would be used in this research, structural, behavioral, and hybrid architecture. There are two parts that can be varied by using three archi-tecture, flip-flop part and input part. In structural architecture, both flip-flop and input part are struc-tural. The code that would be written was gate-ba-sed code. For instance, in flip-flop code, Figure 1 until Figure 4 would be used as a base for the code. Meanwhile, in the input part, Karnaugh Map (K-Map) would be employed as the tools for deter-mining the input of the flip-flop which conformed with the type of the flip-flop. Starting from de-fining the state diagram, thence figuring the state table, and finally finding the combination of the logic gates that would be inserted to the input of the flip-flop. The excitation table to determine the in-put can be seen in Table 1.

In behavioral architecture both part (i.e. flip-flop and input part) would be in behavioral code. The flip-flop code would be written based on the characteristic of the flip-flop, for example in D flip-flop if the clock is rising, then the value of the output would be equal to the value of D (input). Whereas in input part, the value of the output would be depended upon the value of the selector. Firstly, if the value of selector is '00' then the output must be the previous output. Secondly, if the value of selector is '01', the value of output must be the load. Thirdly, if the value of the selector is '10' then the value of the output must be the next bit or I left for bit n. Finally, if the value of the selector is '11', the value of the output must be the previous bit or I right for bit 0. Based upon the selector value, flip-flop type, and output value, the input would be determined and will be more thoroughly elaborated in the experimental setup section.

In hybrid architecture, the architecture of flip-flop code and input code would be differed. This approach is the most reliable and most efficient method, since in the code can be selected the most suitable architecture for each sub-task. In this re-search, behavioral would be utilized in flip-flop co-de, whereas structural would be used

TABLE 1
FLIP FLOP'S EXCITATION TABLE

O(t+1)	D	Operation
0	0	Reset
1	1	Set

O(t+1)	T	Operation
O(t)	0	No change
$\bar{O}(t)$	1	Complement

O(t)	O(t+1)	S	R	Operation
0	0	0	X	No change
0	1	1	0	Set
1	0	0	1	Reset
1	1	X	0	No Change

O(t)	O(t+1)	J	K	Operation
0	0	0	X	No change
0	1	1	X	Set
1	0	X	1	Reset
1	1	X	0	No Change

in input code. The deliberations of these selection were the con-venience of the programming and the error occur-rence consideration.

There are five parameters that would be used in this research to examine the performance of the flip-flop. First parameter is power consumption of the system. Power consumption is very important in designing the FPGA. If the power consumption is large, the system would not be very competitive in the market. In this research, Xilinx XPower Ana-lyzer which is provided by Xilinx ISE Design Suite would be used to estimate the power that consumed by the system. In this application, power consump-tion is divided into two categories, based on the on-chip, and based on supply power. In on-chip power consumption, the energy is absorbed by five differ-rent components, clocks, logic, signal, input/out-put, and leakage. Meanwhile, in the supply power, is absorbed by dynamic power and quiescent pow-er.

The second parameter can be used to investi-gate the performance of the flip-flop is resources necessitated. The resources itself comprise LUT, flip-flop, and slice. This parameter is very impor-tant because it determines the resources would be needed by the system, and thus indirectly deter-mine the price of the system. The more resources required for the similar purpose, the worse the device would be. The third parameter would be found out in this research is memory requirement for system to be executed. As mentioned in the previous section that FPGA only provide limited amount of memory, hence the lesser the memory utilized the desirable it would be. The fourth parameter is the latency of the execution. The latency of the execution is just simply the time needed to complete one operation. In other words, latency is total CPU completion time similar to the criteria explained in the introduction section.

The last parameter that can be assessed to investigate the performance of the flip-flop is effi-ciency of the system. Efficiency is simply defined by the ratio between throughput and used gate (eq-uation(1)). Throughput is the amount of input whi-ch can be processed at certain time, and simply has the unit of MB/s. Throughput can be found by uti-lizing equation(2).

$$efficiency = \frac{throughput}{gate} \quad (1)$$

$$throughput = \frac{memory}{time} \quad (2)$$

Meanwhile, used gate is the number of logic gates that used by the system. The number of gate is not informed by the software unlike resources such as LUT and flip-flop. Therefore, to find the number of gates, method that can be used is to cha-nge the number of LUT and flip-flop into gate number. The multiplication factor of those LUT and flip-flop into gate are summarized in Table 2.

TABLE 2
MULTIPLICATION FACTOR OF RESOURCE INTO GATE

Resource	Gate
LUT	5-6
Flip-flop	7-8

Experimental Setup

Structural architecture

In structural architecture, K-Map would be used to figure out the logic gates and its combination for the input of the flip-flop in each type of flip-flop. To create K-Map, the state table has to be created in advance. In state table step, the value of input (i.e. D, T, S-R, and J-K) would be determined bas-ed upon table I. Even though each flip-flop has si-milar previous and current output, it has different input and conforms to table I. The state table would have 1 bit output and 6 bit output. The output wo-uld be the current output, whereas the input would be from the selector 2 bits, I right, I left, load, and prior output.

The K-Maps input of each flip flop are divi-ded into two groups, RI_LO which stands for I right, load input, and previous output, and the other gro-up is S_0S_1L which stands for selector bit 0, selector bit 1, and I left. From K-Maps, the logic gates arra-ngement could be figured out. The logic gates cir-cuitries that have been simplified and would be us-ed in this research are shown in equation(3) to equ-ation(8).

$$\boldsymbol{D \equiv [\overline{S_0}\overline{S_1}O_0] + [\overline{S_0}S_1I_L] + [S_0\overline{S_1}R] + [S_0S_1L]} \quad (3)$$

$$\boldsymbol{T \equiv [\overline{S_0}\overline{S_1}O_0] + [S_0\overline{S_1}I_L] + [\overline{S_0}S_1R] + [S_0S_1L]} \quad (4)$$

$$\boldsymbol{S \equiv [S_0\overline{S_1}I_L] + [\overline{S_0}S_1R] + [S_0S_1L]} \quad (5)$$

$$\boldsymbol{R \equiv [S_0\overline{S_1}\overline{I_L}] + [\overline{S_0}S_1\overline{R}] + [S_0S_1\overline{L}]} \quad (6)$$

$$\boldsymbol{J \equiv [S_0\overline{S_1}I_L] + [\overline{S_0}S_1R] + [S_0S_1L]} \quad (7)$$

$$\boldsymbol{K \equiv [S_0\overline{S_1}\overline{I_L}] + [\overline{S_0}S_1\overline{R}] + [S_0S_1\overline{L}]} \quad (8)$$

Scenario

The scenario of the research would be conducted in Shift register with parallel load code with four dif-ferent flip-flops (D flip-flop, T flip-flop, S-R flip-flop, and J-K flip-flop). Experiment would be con-ducted on 4, 8, 16, 32, 64, and 128 bit and three ar-chitectural types (i.e. structural,

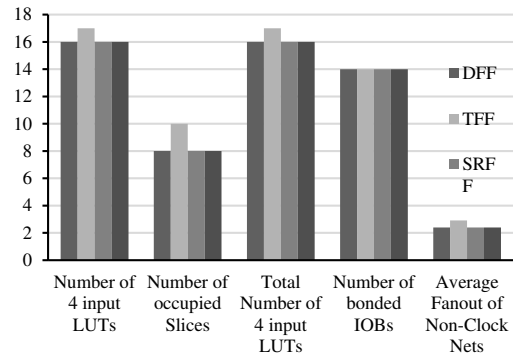

Figure 3. Resource comparison based on flip-flop type.

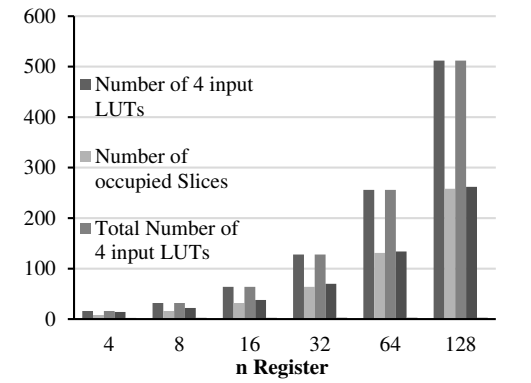

Figure 4. Resource comparison based on the number of register.

behavioral, and hybrid). Finally, the parameters that would be asse-ssed are power consumption, resources required, memory required, latency, and efficiency.

FPGA type

The type of FPGA that would be used in the research is Spartan 3A. Spartan 3A device has flexible power management, leading connectivity platform, abundant, and flexible logic resources. More-over, this type of device has dedicated resources for high-speed DSP applications, precise clock mana-gement up to eight DCMs, and integrated flash RAM memory. Finally, Spartan 3A has large capa-city that would be suitable for the applications that was undertaken in the experiment.

3. Results and Analysis

In this paper, there are five parameters that can be compared to investigate flip-flop's performance. First of all is resource comparison, which means how big of the resources used. Second, memory comparison of the system that conducted. Third is time consumption comparison which means how long it takes to execute the system. Fourth is power comparison, means which system used the lowest power. The last is efficiency comparison, which means how efficient the scheme and which scheme is the most efficient.

In each compared parameter, would be analyzed based on flip-flop types (D flip-flop, T flip-flop, SR flip-flop and JK flip-flop), number of register (n = 4, 8, 16, 32, 64, 128) and VHDL programming architecture (behavioral, structural and hybrid).

First is resource comparison, as shown in Figure 3 there is no significant differences of flip-flop types, all types used almost the same amount of resource. In Figure 4, there are differences of res-ource that used based on number of register. So more register used more resource used too. In Fi-gure 5, each architecture used same amount of res-ource except for structural architecture (two times higher than the other scheme) because in the struc-tural architecture, resource count based on total of gates that used in program. So more complex archi-tecture more complex program would be and more resource is used.

The second parameter is the memory comparison, similar to the first parameter, there are three indicators that were compared on this parameter. The first is based on the type of the flip-flop (struc-tural architecture with 4 registers). Shown in figure 6, the difference in the amount of used

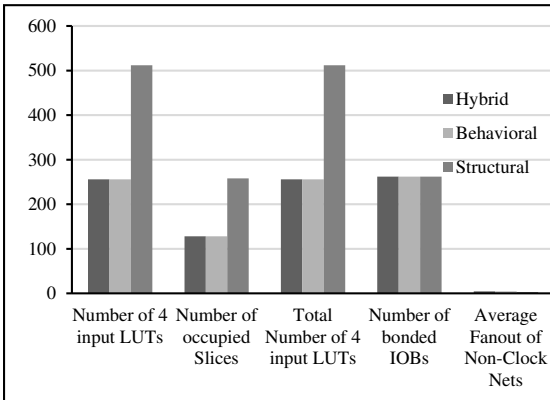

Figure 5. Resource comparison based on architecture.

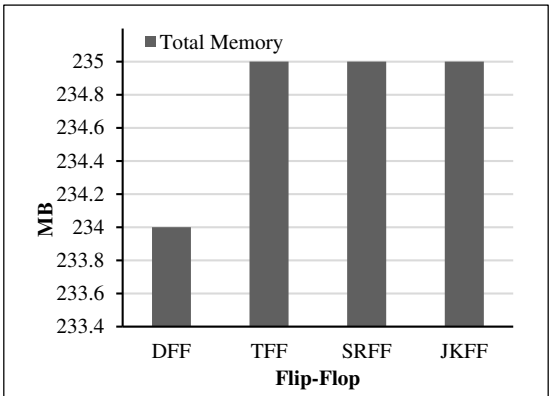

Figure 6. Memory comparison based on flip-flop type.

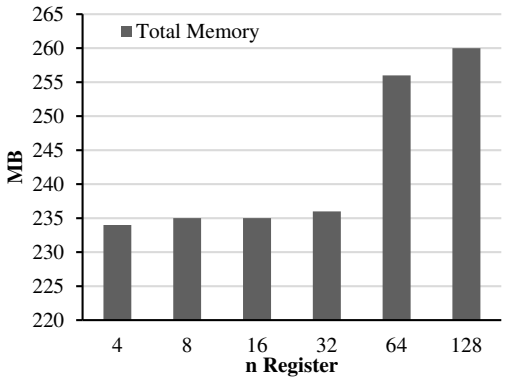

Figure 7. Memory comparison based on the number of register.

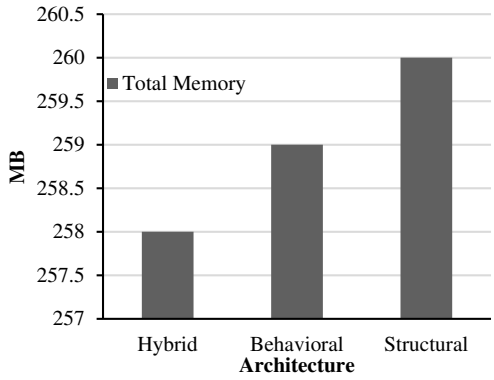

Figure 8. Memory comparison based on architecture.

memory is not too significant, each type of flip-flop have the same amount of used memory that is between 234 MB to 235 MB.

The second indicator is based on the number of used register which is shown in Figure 4. The greater the number of registers, the larger the mem-ory is used. It is also shown by Figure 7, when the number of registers are 4 and used memory is 234 MB.

Then if the number of register rise to 32 the used memory is 236 MB and so on. This difference appeared because when the number of registers increase, automatically the greater the amount of program computation and then the greater the me-mory that is used when computing the larger prog-ram.

Thereupon, when compared based on VHDL programming architecture, as shown in figure 8, hybrid architecture (a combination of behavioral and structural architecture) has the amount of used memory is 258 MB (flip-flop type: DFF; the number of registers, n = 128). This result is the smallest among the other two architectures (behavioral arc-hitecture is 259 MB and structural architecture is 260 MB). It happened because hybrid architecture made simple representation for such a system, it is different from the structural architecture which mo-re complex the systems, more complex the pro-gram and the greater used memory. Beside hybrid architecture, D flip flop tended to consume the less-er amount of memory, mainly because of the sim-plicity of the flip-flop structure despite of more complex input structure. In addition, in another flip-flop there is "reset" function to ensure the exe-cution of the program without failure.

On the third parameter, it has significant difference in comparison based on flip-flop type, the number of registers, and VHDL programming architecture. In figure 9, it is indicated that the D flip-flop has the shortest computation time compared to the other types of flip-flop that is equal to 15 seconds for the REAL time and CPU time. The slo-west computation time owned by T flip-flop with 20 seconds then the JK flip-flop with 19 seconds meanwhile SR flip-flop has the second fastest computation time with 16 seconds. Even though D flip flop has more complex input structure, the completion time of this type was still the most rapid due to the simplicity of the flip-flop structure itself. More-over, the other flip-flops had to be reset at the beginning to evade the "undefined", thus it would take a longer time (this

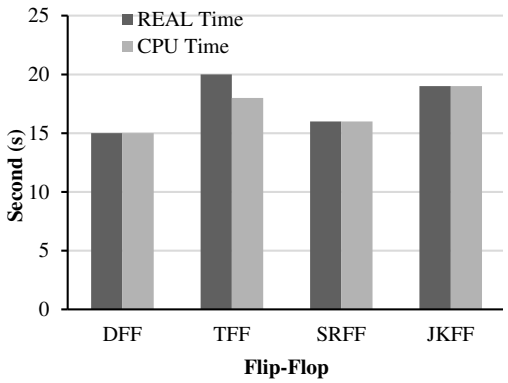

Figure 9. Computation time comparison based on flip-flop type.

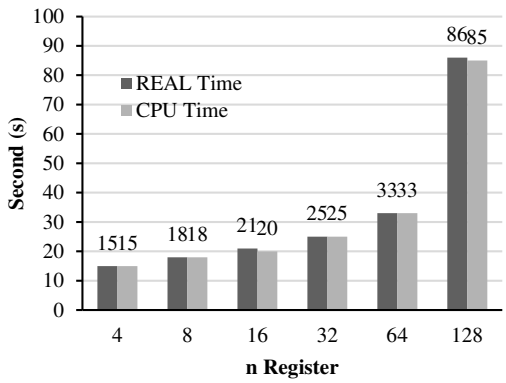

Figure 10. Computation time comparison based on the number of register.

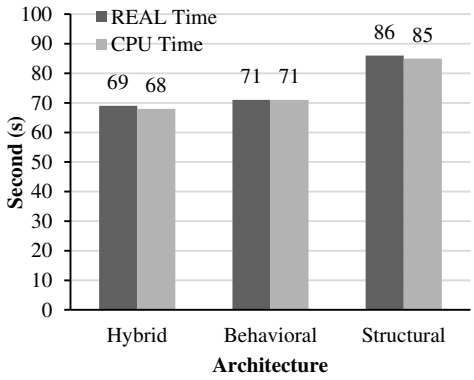

Figure 11. Time comparison based on architecture.

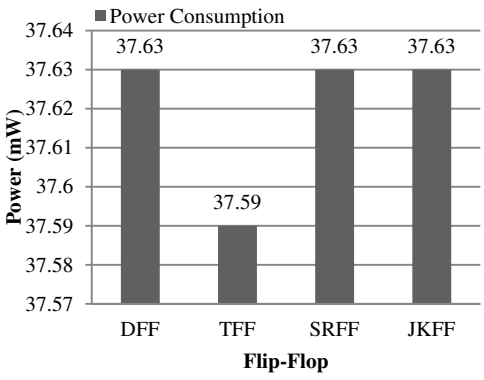

Figure 12. Power consumption comparison based on flip-flop type.

conforms to the previous explanation).

In the Figure 10, as discussed before that the greater the number of registers, the greater of resource or memory is used. This also applies to the computation time, the greater the number of registers the longer the computation time.

In the Figure 11, hybrid architecture shows the best result in computation time comparison. It is about 3 to 17 seconds faster than the other two architectures for CPU time completion and about 2 to 17 seconds faster than two other architecture for REAL time completion.

In the Figure 12, power consumption comparison based on the type of flip-flop (architecture: structural; n=4) is obtained, the power consumption for each of the flip-flop is not dramatically dif-ferent. For the D flip-flop the total of power con-sumption is 37.63 mW. This is similar with SR flip-flop and JK flip-flop. Meanwhile, T flip-flop has shown a lower power consumption, approximately 37.59 mW. In fact, there is no reasonable explana-tion for the smaller energy consumption of T flip-flop. The only possible elaboration is the deviation of the experiment itself.

The second is a comparison of power consumption based on the number of registers (flip-flop type: D flip-flop; architecture: structural) is obtain-ned, the power consumption for n = 4 is 37.63 mW while n = 8 is 37.99 mW, next when n = 16 the power consumption is 37.66 mW, then n = 32 the power consumption is 38.09 mW and for n = 64 and 128, the power consumption is 39.29 mW and 41.18 mW respectively. This result is shown in Fi-gure 13 that the greater the number of registers used, the greater of the power consumption would be. Although, when n = 8 has higher number of register than n = 16. The phenomenon happened because there are quiescent of power whose value is uncertain.

The last one is power consumption comparison based VHDL programming architecture shown in Figure 14 (flip-flop type = D flip-flop; n = 128). The power consumption for the hybrid architecture is 39.89 mW whereas for behavioral architecture, the power consumption is 40.71 mW. The greatest power consumption absorbed by structural archi-tecture is about 41.18 mW, this results influenced by the amount of resource, memory and computa-tion time of structural architecture is larger than the other two architecture (hybrid and behavioral). The last parameter is the ratio of efficiency, each system

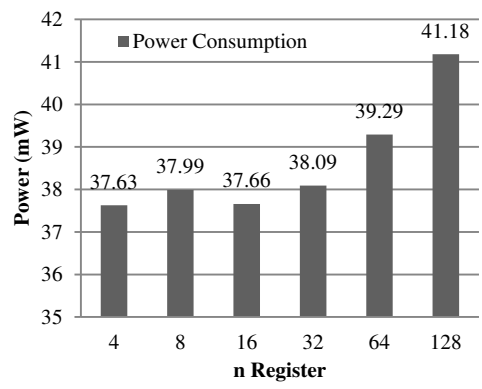

Figure 13. Power consumption comparison based on the number of register.

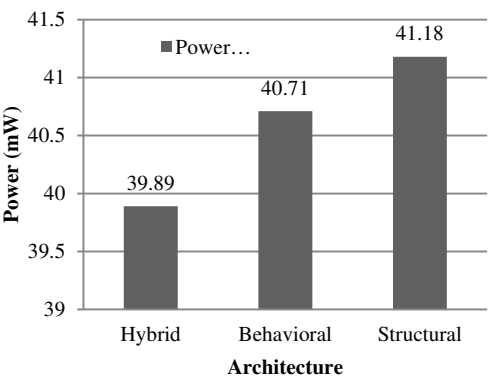

Figure 14. Power Consumption comparison based on architecture.

TABLE 3
EFFICIENCY COMPARISON

		# Gates	Total Execution Time (s)	Total Memory (MB)	Throughput (MB/s)	Efficiency
Number of Register (DFF; Structural) :	4	96	15	234	15.6	0.1625
	8	192	18	235	13.05556	0.067998
	16	384	20.5	235	11.46341	0.029853
	32	768	25	236	9.44	0.012292
	64	1536	33	256	7.757576	0.005051
	128	3072	85.5	260	3.040936	0.00099
Type of Flip-Flop (n=4; structural):	DFF	96	15	234	15.6	0.1625
	TFF	102	19	235	12.36842	0.121259
	SRFF	96	16	235	14.6875	0.152995
	JKFF	96	19	235	12.36842	0.128838
Scheme (DFF; n=128):	Hybrid	1536	68.5	258	3.766423	0.002452
	Behavioural	1536	71	259	3.647887	0.002375
	Structural	3072	85.5	260	3.040936	0.00099

can be considered good when the efficiency of the system high. Efficiency as described in the previous chapter is a comparison between the through-put and the number of gates used. Getting closer to a value of 1, the system is considered efficient. On the Table 3 can be seen a comparison of efficiency based on number of registers, then flip-flop type and VHDL programming architecture. As shown in number of registers, more efficient system is the system with the least number of registers, the effi-ciency is about 0.1625. Then, when viewed from the flip-flop type then more efficient system is the system with the D flip-flop the efficiency is about 0.1625 and the last is shown by the VHDL pro-gramming architecture more efficient system is the system with a hybrid architecture (a combination of behavioral and structural) the efficiency is about 0.002452.

4. Conclusion

In this paper, we investigated the performance of the flip-flop based on differences in the type, num-ber of registers and architecture used in shift regis-ter with parallel load applications. Comparison res-ults showed that type of flip-flop is not very influ-ential in resource comparison, while for memory comparison, D flip-flop showed the best results (used minimal memory). The same trend was sho-wn for computation time, power consumption and system efficiency. Broadly speaking, the D flip-fl-op showed the best performance for each of these parameters. Then for comparison based on the nu-mber of registers, the results obtained for each pa-rameter that conducted the greater number of the register the more resource and memory used, more computation time would be, then the power consu-mption increase and finally smaller the efficiency of the system. Finally for architecture comparison that hybrid architecture (a combination of behave-oral and structural architecture) showed the best performance compared with other architecture.

References

[1] Paanshul Dobriyal, Karna Sharma, Manan Sethi, Geetanjali Sharma, "A High Performance D-Flip Flop Design with Low Power Clocking System using MTCMOS Technique" *In 3rd IEEE International Advance Com-puting Conference,* pp. 1524-1528, 2013.

[2] Nidhi Gupta, "Clock Power Analysis of Low Power Clock Gated Arithmetic Logic Unit on Different FPGA" *In 6th International Confe-rence on Computational Intelligence and Co-mmunication Networks,* pp. 913-916, 2014.

[3] Hazem A. Ahmed, Hamed Salah, Tallal Elshabrawy, Hossam A. H. Fahmy, "Low Energy High Speed Reed-Solomon Decoder Using Two Parallel Modified Evluator Inversionless Berlekamp-Massey" *In 19th IEEE Interna-tional Conference on Electronics, Circuits, and Systems (ICECS),* pp. 396-399, 2012.

[4] Sodyan Redif and Server Kasap, "Novel Reconfigurable Hardware Architecture for Polynomial Matrix Multiplications" *In IEEE Transactions on Very Large Scale Integration (VLSI),* pp. 454-465, 2015.

[5] Priya Mathew, Deepak Kushwaha, Vivian Desalphine and A. David Selvakumar, "Hard-ware Implementation of NB PHY

Transceiver of IEEE 802.15.6 WBAN on FPGA" *In In-ternational Conference on Medical Imaging, m-Health and Emerging Communication Sys-tems (MedCom),* pp. 64-71, 2014.

[6] Najeem Lawal, Benny Thörnberg, and Mattias O' Nils, "Architecture driven memory al-location for FPGA Based Real-Time Video Processing Systems" *In IEEE 7th Southern Conference on Programmable Logic,* pp. 143-148, 2011.

[7] Amit Kumar Panda, Shahbaz Sarik, and Abhishek Awasthi, "FPGA Implementation of Encoder for (15, k) Binary BCH Code Using VHDL and Performance Comparison for Mu-ltiple Error Correction Control" *In Interna-tional Conference on Communication Syst-ems and Network Technologies,* pp. 780-784, 2012.

[8] Philippe Dondon, Julien Carvalho, Rémi Gar-dere, Paul Lahalle, Georgi Tsenov and Valeri Mladenov, "Implementation of a Feed-for-ward Artificial Neural Network in VHDL on FPGA" *In 12th Symposium of Neural Network Applications in Electrical Engineering (NE-UREL),* pp. 37-40, 2014.